REMARKS

The Office Action of September 3, 2002 rejected claims 1-13. Specifically, the Office Action rejected claims 1-4 and 7-13 under 35 U.S.C. § 102(e) as being anticipated by "Image Acquisition Using TWAIN" by Craig A. Lindley ("Lindley"). Claims 5 and 6 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Lindley. Applicants respectfully request favorable reconsideration of the pending claims in view of the amendments and the remarks herein.

As defined in the preamble of claim 1, the invention of claim 1 is directed to "[a] method for representing to an application the characteristics of an underlying connection-oriented device over known application-level interfaces and allowing an application to take advantage of a connection-oriented I/O subsystem having an integrating component over the known application-level interfaces." Moreover, as specified in claim1, first and second known application-level interfaces associated with the integrating component are used to represent to the application the connection control characteristics and the data and data control characteristics, respectively, of the underlying connection-oriented device. Thus, the method of claim 1 is clearly recited in the context of a connection-oriented device.

Claim 1 has been amended by further specifying that the connection control characteristics represented to the application over the first known application-level interface are "related to the manner in which the connection-oriented device makes a connection for sending and receiving network data over a network".

In contrast to the connection-oriented device of claim 1 and the acts by which the integrating component represents the associated connection control characteristics and data and data control characteristics to the application, Lindley describes a system for providing a uniform

interface between software and image-capturing hardware. (Lindley, page 76, column 1). The hardware described by Lindley includes scanners, digital cameras and the like. The hardware devices of Lindley and the requirements for operation thereof are significantly different from the connection-oriented devices of claim 1. Thus, for at least this reason, Lindley fails to disclose all of the elements of claim 1.

The scanners and digital cameras of Lindley do not have the connection control requirements of connection-oriented devices. Thus, Lindley fails to disclose representing to an application "the connection control characteristics of an underlying connection-oriented device" or the "data and data control characteristics of the underlying connection-oriented device." The element of claim 1 relating to the representation of the "connection control characteristics . . . of the connection-oriented device" clearly distinguishes from Lindley, since the hardware (e.g., scanners, digital cameras) that is used with the interface disclosed by Lindley does not have connection control characteristics. For this additional reason, claim 1 distinguishes from Lindley.

In addition to the fundamental distinction between the connection-oriented devices of claim 1 and the hardware of Lindley, the Lindley reference fails to disclose certain elements that are clearly recited in claim 1. For example, claim 1 recites a "first known application-level interface" and a "second known application-level interface" associated with the integrating component. According to the Office Action of September 3, 2002, the "Source Manager" of Lindley is analogous to the integrating component of claim 1. Lindley describes the Source Manager as being a "go-between" for the application and a Source. (Lindley, page 76, column 2). Figure 1 of Lindley further illustrates the Source Manager as being a go-between for the application and a Source.

Assuming, *arguendo*, that the Source Manager of Lindley is analogous to the integrating component of claim 1, Applicants respectfully submit that the Office Action misconstrues the fundamental distinction between an interface and a call to the interface. Claim 1 defines two interfaces (i.e., the first and second known application-level interfaces) associated with the integrating component. In contrast, Lindley clearly specifies that the Source Manager has a "single point of entry," namely, *DSM_Entry*, and that the "application program controls the acquisition process by making calls" to this single point of entry. (Lindley, page 76, column 3, emphasis added). Thus, the Source Manager go-between component has only a single interface (i.e., the single point of entry, *DSM_Entry*) to which all calls by the application are made. Because claim 1 recites two interfaces associated with the integrating component, claim 1 distinguishes from Lindley.

The Office Action asserts that the "Select Source" operation represents a first interface and that the "Acquire" or "Set Resolution" operation represents a second interface. However, a careful review of Lindley reveals that these operations are not interfaces to the Source Manager go-between component, but are instead "menu selections" or "operations" that enable the user to control the acquisition of images. (Lindley, page 76, column 3). Regardless of the number of menu selections or operations that are presented to a user according to Lindley, the fact remains that Lindley describes the Source Manager as having only one interface (i.e., the single point of entry, *DSM_Entry*) and that all calls made by the application program are directed to this single interface. Thus, for these reasons, claim 1 distinguishes from Lindley.

Independent claims 5 and 10, include limitations similar to those of claim 1 referenced above, and these claims likewise distinguish from Lindley. In addition, Applicants respectfully submit that many of the pending claims include other limitations that further distinguish them

from Lindley. The Office Action of September 3, 2002 failed to include any allegation that many of these limitations were disclosed by Lindley. For instance, although the Office Action rejected claims 2-5, 8, 11 and 12, these claims include numerous elements that were not referenced in the Office Action. Thus, Applicants respectfully submit that no *prima facie* case of obviousness has been made with respect to these claims, and that these claims are in condition for allowance for this additional reason.

In particular, the Office Action fails to identify any portion of Lindley that discloses numerous elements of claim 5. For example, claim 5 recites "a data transport capable of communication with an application," "providing facility for associating the connection with the data transport", and "causing redirection of data and data control information from the connection through the proxy client component to a designated data transport designated in one of the abstract connection control commands." Applicants respectfully submit that the mechanism for providing interoperability between graphics-supporting software and image-gathering hardware of Lindley does not operate in the manner defined in claim 5 and, specifically, does not include the foregoing elements that are recited in claim 5.

Applicants note that, in rejecting claim 5, the Office Action states that it would have been well known to use "proxy mechanisms and a local handle or identifier to the proxy object." Based on this statement, it appears that the Office Action takes official notice of these elements. Applicants respectfully submit that even if these elements were well known in the art, it would not have been obvious to combine them with the methods disclosed by Lindley. Applicants respectfully traverse this official notice.

Attached hereto is a marked-up version of the changes made to the previous version of the specification and claims by this amendment. The attached pages are captioned "VERSION WITH MARKINGS TO SHOW CHANGES MADE."

For the foregoing reasons, Applicants submit that the pending claims are in condition for allowance and courteously request favorable action. If there are any outstanding issues that could be resolved by telephone, the Examiner is invited to contact the undersigned attorney.

Dated this 31 day of January, 2003.

Respectfully submitted,

R. Burns Israelsen

R. BURNS ISRAELSEN
Attorney for Applicant
Registration No. 42,685

WORKMAN, NYDEGGER & SEELEY
1000 Eagle Gate Tower
60 East South Temple
Salt Lake City, Utah 84111
Telephone: (801) 533-9800
Facsimile: (801) 328-1707

RBI:blw
W:\13768\73\BLM0000000735V001.doc

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE CLAIMS

Claims 1, 5, 7 and 10 have been amended as follows:

1. (Thrice Amended) A method for representing to an application the characteristics of an underlying connection-oriented device over known application-level interfaces and allowing an application to take advantage of a connection-oriented I/O subsystem having an integrating component over the known application-level interfaces and without requiring the application programmer to program directly to the integrating component, the method comprising:

representing to an application, over a first known application-level interface associated with the integrating component, the connection control characteristics of the underlying connection-oriented device related to the manner in which the connection-oriented device makes a connection for sending and receiving network data over a network, wherein the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device;

representing to the application, over a second known application-level interface associated with the integrating component, the data and data control characteristics of the underlying connection-oriented device;

receiving, by the integrating component, a command from the application over the first known application-level interface;

receiving, by the integrating component, a command from the application over the second known application-level interface; and

by the connection-oriented device driver, interacting with the integrating component in order to execute said received commands so that the application may take advantage of the connection-oriented I/O subsystem and use the connection-oriented device using the known application-level interfaces and without requiring the application programmer to program to an interface of the connection-oriented device driver.

5. (Twice Amended) A connection-oriented driver subsystem where connection control information is communicated to an application through a connection interface while data

and data control information is communicated through a transport interface, the driver subsystem comprising:

a connection-oriented device driver controlling a connection-oriented hardware device;

a data transport capable of communication with an application;

an integrating component that interfaces with the connection-oriented device driver and the data transport, said connection-oriented device driver and said data transport serving as clients to said integrating component, wherein said integrating component is positioned between the application and the connection-oriented device driver, said integrating component:

providing an abstracted connection interface that is available to a client that allows the client to create a connection with a desired location using the connection-oriented hardware device; and

providing facility for associating the connection with the data transport, thereby allowing the client to send and receive data and data control information over the connection; and

a proxy client component that interfaces with the connection interface and the transport interface of the integrating component as a client, said proxy client component:

receiving abstract connection creation commands and abstract connection control commands from the application and implementing said commands through use of the connection interface to create and manage the connection;

causing redirection of data and data control information from the connection through the proxy client component to a designated data transport designated in one of the abstract connection control commands; and

returning to the application, in response to a previously received connection control command, an identifier to be used by the application for receiving data and data control information from the designated data transport so that the connection control information is communicated to

the application through the proxy client component while the data and data control information is communicated to the application through the designated data transport.

7. (Thrice Amended) A computer program product for interacting with known application-level interfaces and an integrating component of a connection-oriented I/O subsystem in order to represent the characteristics of an underlying connection-oriented device to an application and allow an application to take advantage of the connection-oriented I/O subsystem over the known application-level interfaces without requiring the application programmer to program to a new interface, said computer program product comprising:

a computer-readable medium; and

computer-executable instructions carried on said computer-readable medium for performing the steps of:

representing to an application, over a first known application-level interface associated with the integrating component, the connection control characteristics of the underlying connection-oriented device [devices to an application over a first known application level interface associated with the integrating component] related to the manner in which the connection-oriented device makes a connection for sending and receiving network data over a network, wherein the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device;

representing the data and data control characteristics of the underlying connection-oriented device [devices] to the application over a second known application level interface associated with the integrating component;

receiving, by the integrating component, a command from the application over the first known application-level interface;

receiving, by the integrating component, a command from the application in the second known application-level interface; and

by the connection-oriented device driver, interacting with the integrating component to execute said received commands.

10. (Twice Amended) A method for representing to an application the characteristics of an underlying connection-oriented device over known application-level interfaces and allowing an application to take advantage of a connection-oriented I/O subsystem having an integrating component over the known application-level interfaces and without requiring the application programmer to program directly to the integrating component, the method comprising:

separating connection control characteristics from data and data control characteristics received from an underlying connection-oriented device;

representing to an application, over a first known application-level interface associated with the integrating component, the connection control characteristics of the underlying connection-oriented device related to the manner in which the connection-oriented device makes a connection for sending and receiving network data over a network, wherein the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device;

representing to the application, over a second known application-level interface associated with the integrating component, the data and data control characteristics of the underlying connection-oriented device;

receiving, by the integrating component, a command from the application over the first known application-level interface;

receiving, by the integrating component, a command from the application over the second known application-level interface; and

by the connection-oriented device driver, interacting with the integrating component in order to execute said received commands so that the application may take advantage of the connection-oriented I/O subsystem and use the connection-oriented device using the known application-level interfaces and without requiring the application programmer to program to an interface of the connection-oriented device driver.